Network-on-Chip Router Design with Buffer-Stealing

Wan-Ting Su, Jih-Sheng Shen*, Pao-Ann Hsiung

ASP-DAC 2011

Embedded Systems Laboratory
Department of Computer Science & Information Engineering
National Chung Cheng University

Outline

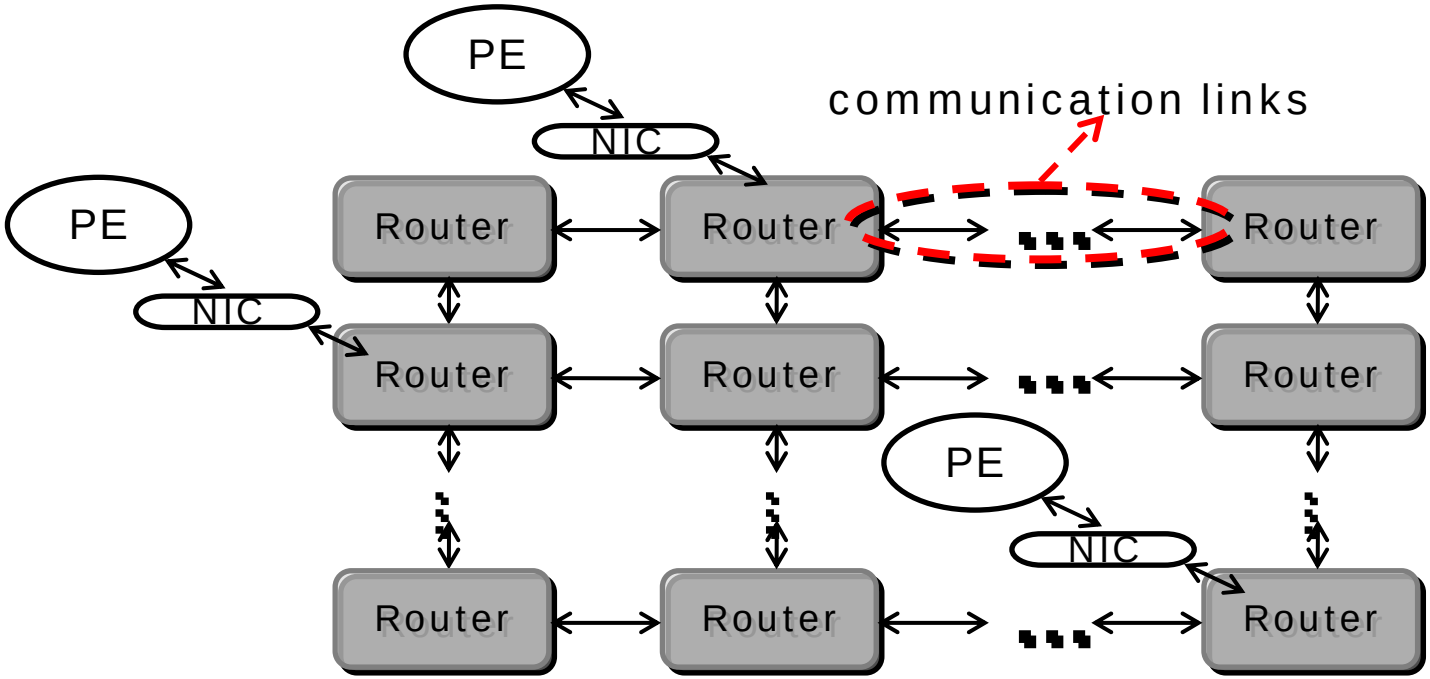
- Introduction
- Motivation and Objective
- Buffer Stealing Design
- Experiments
- Conclusions

Outline

- **Introduction**
- Motivation and Objective
- Buffer Stealing Design
- Experiments
- Conclusions

Network-on-Chip

- A basic Network-on-Chip (NoC) architecture
  - Routers,
  - Communication links,
  - Network-Interface Component (NIC)



NoC Buffer

- The **use of buffering**
  - Wait for routing decisions
  - Compete for the same output channel
  - No buffer space in next hop router

- The **utilization of input buffers** directly influence
  - NoC congestions
  - Throughput
  - Packet latency

Outline

- Introduction
- **Motivation and Objective**
- Buffer Stealing Design
- Experiments
- Conclusions

Motivation

- **Increase** the buffer size (or infinite buffer)
  - Reduce NoC congestions and packet latency
  - Enhance throughput

- However, the problems of infinite buffer are
  - **High hardware resource overhead**
  - **Large energy consumers**
    - 64% of the total router leakage power [32]
    - More dynamic energy consumption [33]

[32] W. Hangsheng, L. S. Peh, and S. Malik, "Power-driven design of router microarchitectures in on-chip networks," in *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture* (MICRO), pp. 105-116, 2003.
[33] T. T. Ye, L. Benini, and G. De Micheli, "Analysis of power consumption on switch fabrics in network routers," in *Proceedings of the 39th Design Automation Conference* (DAC), pp. 524-529, 2002.

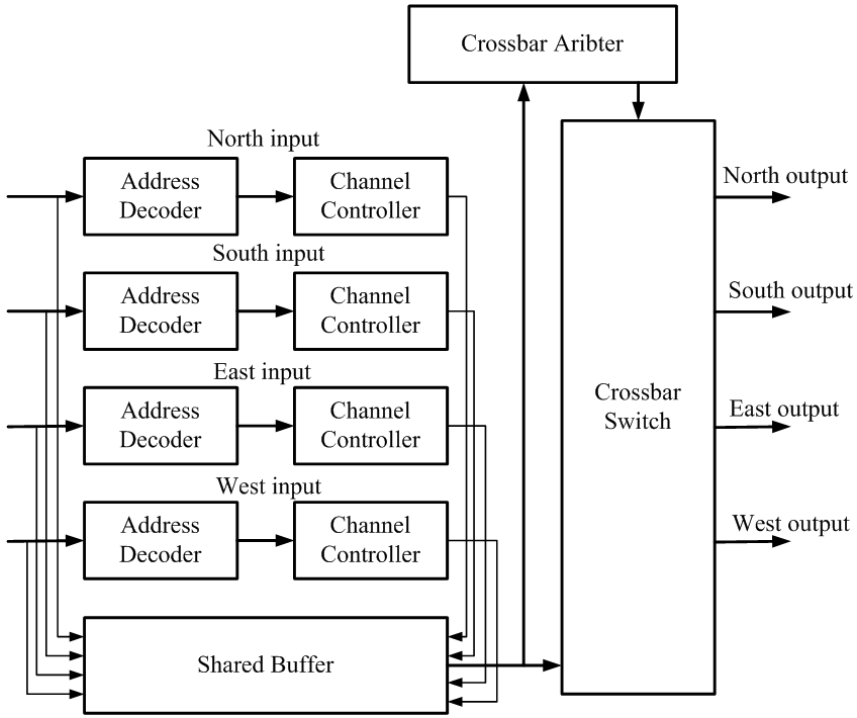
Related Work: Virtual Channel Method

Buffer-sharing methods	Lai [7] (DAC'08)	Liu [4] (CSS'07) Liu [9] (MWSCAS'06)	Neishaburi [11] (GLSVLSI'09)	Our method (BS)
Simulation environment	Cycle-accurate simulator	Flexsim 1.2	RTL, VHDL	RTL, VHDL Modelsim SE 5.8d
Routing algorithm	Dimension routing	Adaptive routing	XY-YX routing	XY routing
Traffic pattern	1) Uniform 2) Hotspot	Uniform	Uniform Transpose	Burst
No. of virtual channel	2-10 VCs (32 bits buffers)	2-8 VCs for one physical channel (one VC = 4 flits)	4-16 VCs (64 bits buffers)	No VC
Performance enhancement	1) Throughput : 8.3% increase 2) Latency : 19.6% decrease	Throughput : • 2% than SAMQ • 1% than DAMQall	Latency : • 7.1% decrease in uniform • 3.5% decrease in transpose	1) Throughput : 40% increase (23.47% in average) 2) Latency : 22.46% decrease (10.17% in average)
Compare to	1) 2 VCs router 2) 4 VCs router	1) SAMQ 2) DAMQall	4 VCs router (one VC = 16 flits)	Extended original buffer (with the same additional hardware overhead)

Virtual channel buffers require up to nearly **50% of area** and account for **64% of leakage power** in a router implemented under the 70 nm CMOS technology.

Related Work: Central Buffer Sharing Method

- Central buffer-sharing method is proposed [17, 18]
  - All ports share a central buffer
  - Area overheads
    - Control complexity
    - Register file



[17] P.-T. Huang and W. Hwang, “2-level FIFO architecture design for switch fabrics in network-on-chip,” in Proceedings of the International Symposium on Circuits and Systems. IEEE, May 2006, pp. 4863–4866.
[18] L.-F. Leung and C.-Y. Tsui, “Optimal link scheduling on improving best-effort and guaranteed services performance in network-on-chip systems,” in Proceedings of the 43rd Design Automation Conference. ACM, July 2006, pp. 833–838.

Objective

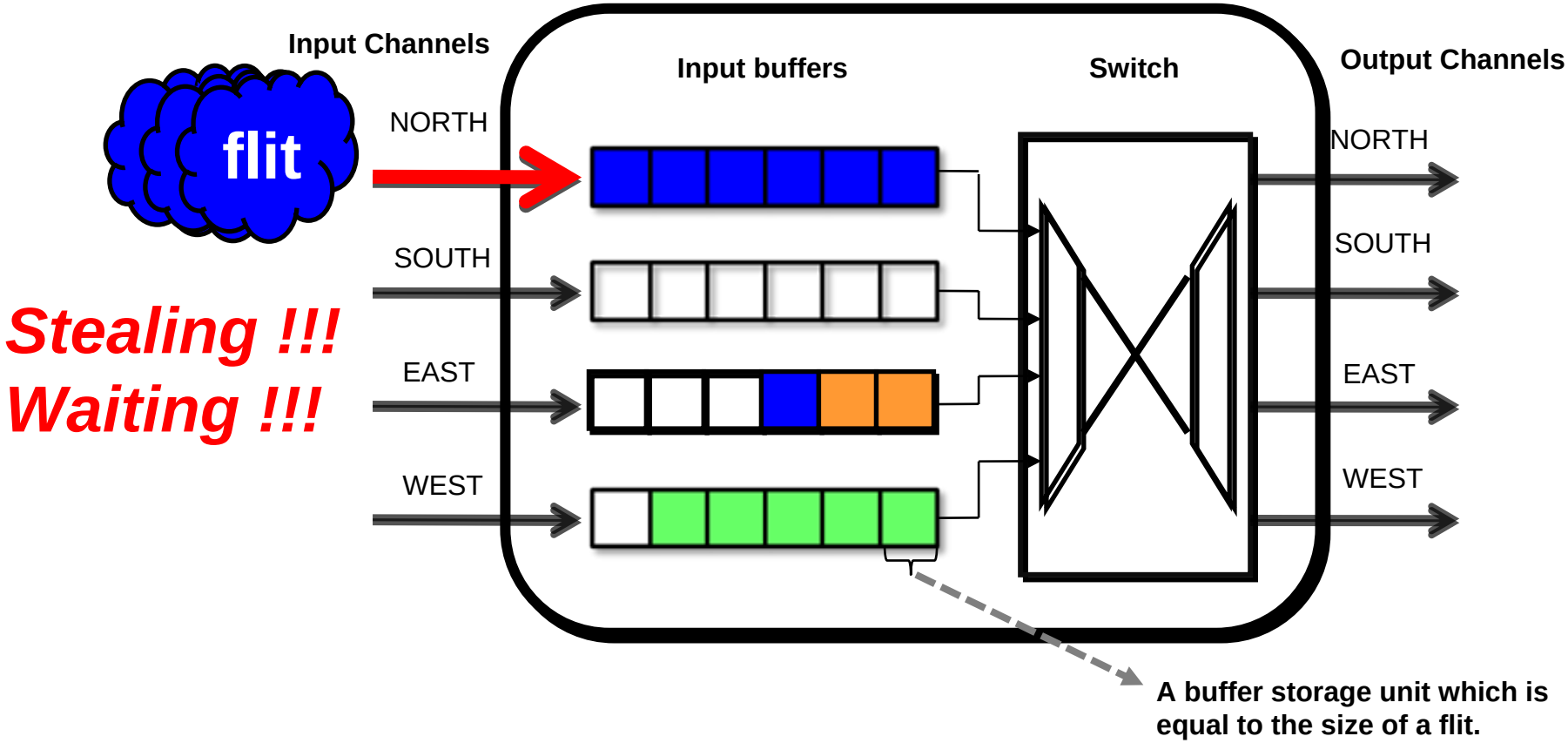
Buffer Stealing (BS) mechanism

- Utilize at **runtime** the **free input buffers from other input channels**
  - Instead of increasing the buffer size at design time.

- Advantages :
  1) Congestion reduction
  2) Throughput Enhancement
  3) Efficient buffer utilization
  4) Low resource overhead

The concept of BS can be easily generalized to routers with VCs;
➢ Consider the hardware overhead and power drawback incurred by VC-routers

An Example of Buffer Stealing



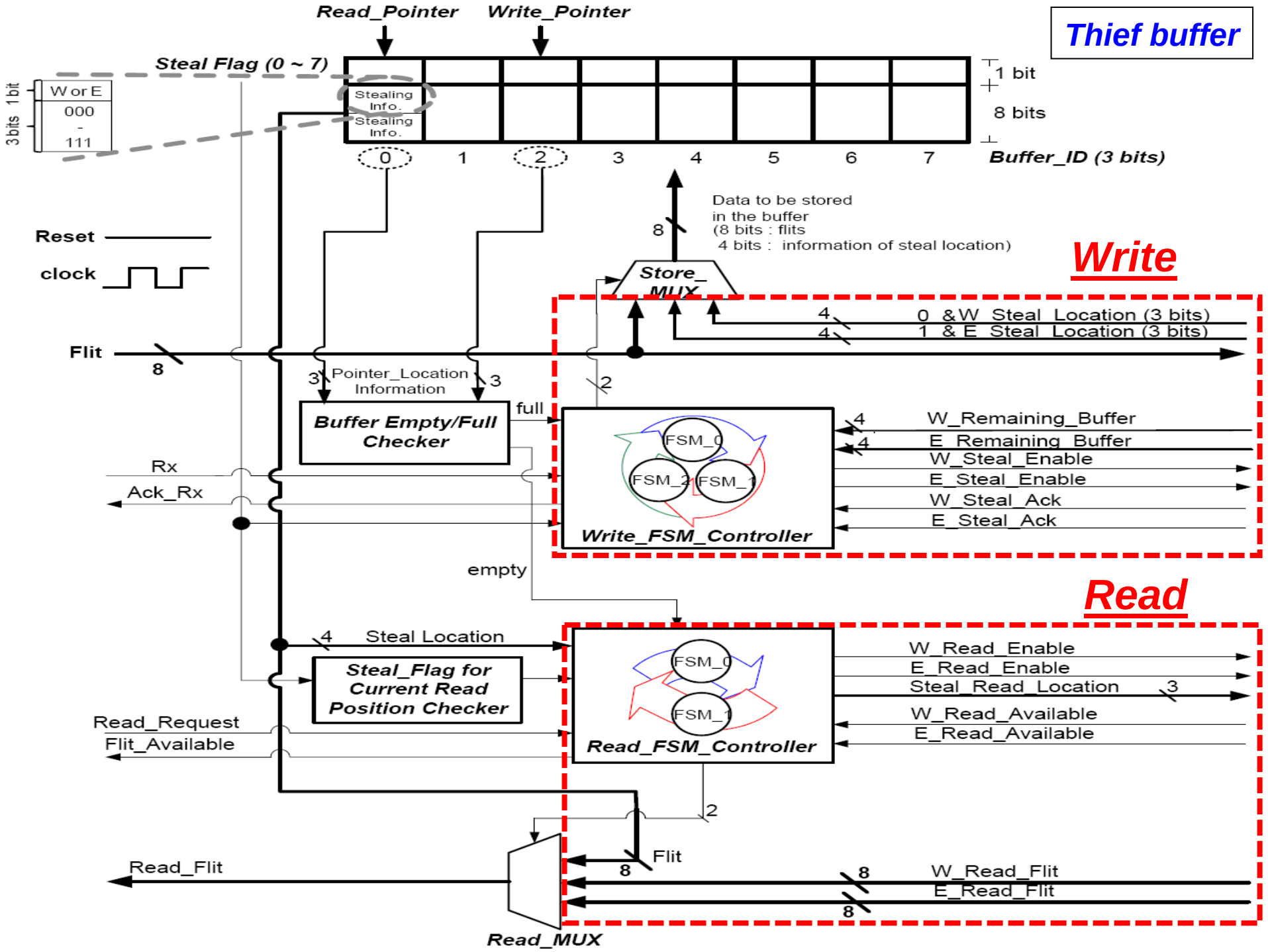
Outline

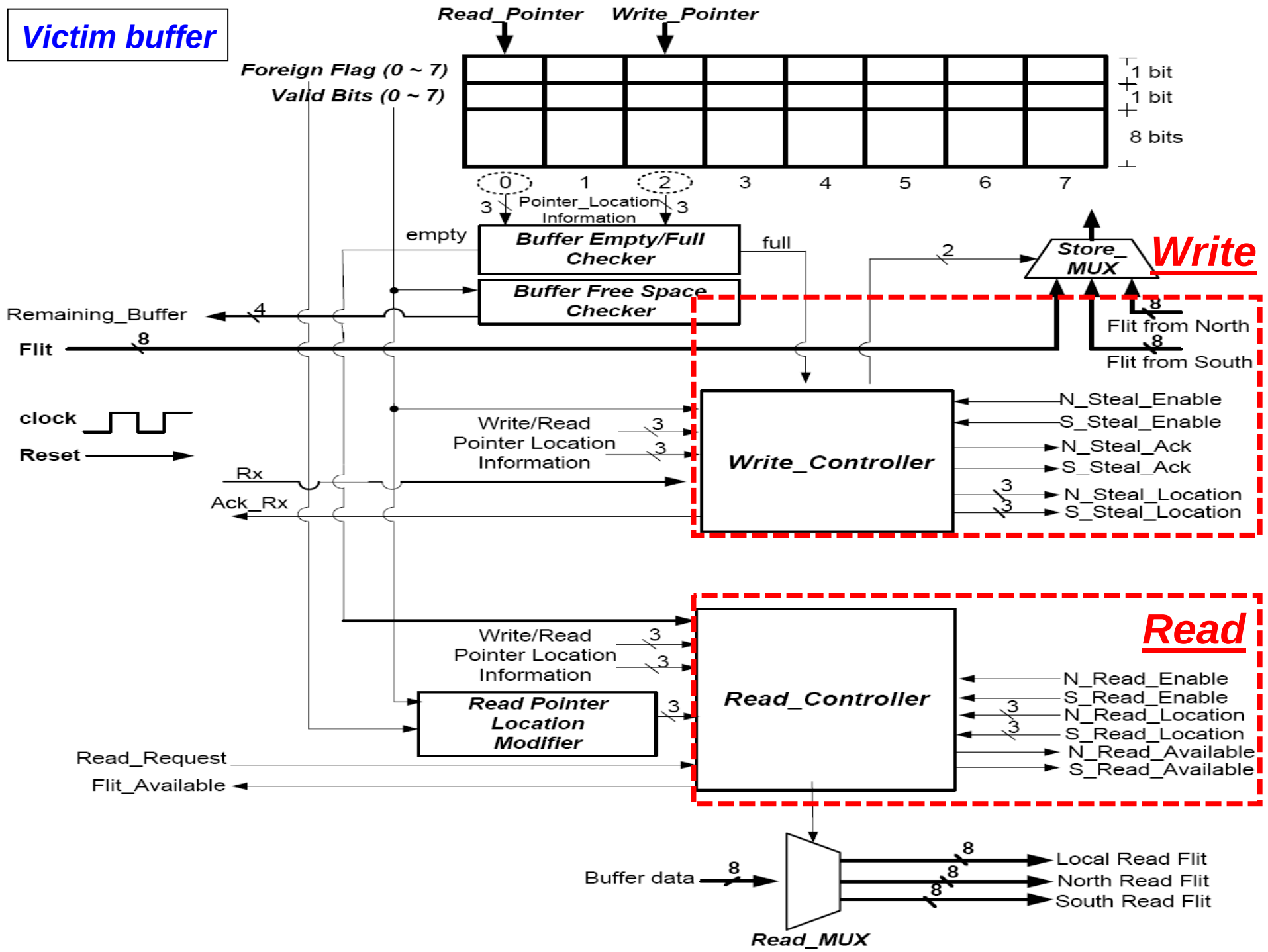
- Introduction
- Motivation and Objective
- **Buffer Stealing Design**
- Experiments
- Conclusions

Circuit Designs for Buffer Stealing

- **Thief buffer design**
  - A thief buffer is a buffer that steals the buffer space of other channels when its free buffer space is not enough to store incoming flits.

- **Victim buffer design**
  - A victim buffer is a buffer whose free space can be stolen by a thief buffer.


Thief buffer
Read_Pointer
Write_Pointer
Steal Flag (0 ~ 7)
3bits 1bit
W or E
000
-
111
Stealing Info.
Stealing Info.
1 bit
8 bits
0 1 2 3 4 5 6 7
Buffer_ID (3 bits)
Reset
clock
Flit
8
Data to be stored in the buffer
(8 bits : flits
4 bits : information of steal location)
Store_ MUX
Write
4 0 &W Steal Location (3 bits)
4 1 & E Steal Location (3 bits)
3 Pointer_Location Information 3
Buffer Empty/Full Checker
full
2
Rx
Ack_Rx
FSM_0
FSM_2
FSM_1
Write_FSM_Controller
4 W_Remaining_Buffer
4 E_Remaining_Buffer
W_Steal_Enable
E_Steal_Enable
W_Steal_Ack
E_Steal_Ack
empty
Read
4 Steal Location
Steal_Flag for Current Read Position Checker
Read_Request
Flit_Available
FSM_0
FSM_1
Read_FSM_Controller
W_Read_Enable
E_Read_Enable
Steal_Read_Location 3
W_Read_Available
E_Read_Available
2
Flit
8
Read_Flit
8 W_Read_Flit
8 E_Read_Flit
Read_MUX


Victim buffer
Read_Pointer
Write_Pointer
Foreign Flag (0 ~ 7)
Valid Bits (0 ~ 7)
1 bit
1 bit
8 bits
0
1
2
3
4
5
6
7
3
Pointer_Location Information
3
empty
Buffer Empty/Full Checker
full
Buffer Free Space Checker
2
Store_ MUX
Write
Remaining_Buffer
4
Flit
8
8
Flit from North
8
Flit from South
clock
Reset
Write/Read Pointer Location Information
3
3
Write_Controller
N_Steal_Enable
S_Steal_Enable
N_Steal_Ack
S_Steal_Ack
3
3
N_Steal_Location
S_Steal_Location
Rx
Ack_Rx
Read
Write/Read Pointer Location Information
3
3
Read Pointer Location Modifier
3
Read_Controller
N_Read_Enable
S_Read_Enable
3
3
N_Read_Location
S_Read_Location
N_Read_Available
S_Read_Available
Read_Request
Flit_Available
Buffer data
8
8
8
8
Local Read Flit
North Read Flit
South Read Flit
Read_MUX

Outline

- Introduction
- Motivation and Objective
- Buffer Stealing Design
- **Experiments**
- Conclusions

System Architecture

- NoC parameters used in this work :
  - A 64-bit 5-input-buffer router

	The selected NoC parameters	Advantages
Arbitration Logic	Round-Robin Algorithm	1. All input requests will be **eventually granted** 2. Prevent **starvation**
Switching Method	Wormhole Switching	1. **Minimal buffering** requirement 2. **Low-latency** communication
Routing Algorithm	X-Y Routing Algorithm	1. **Minimize** the **area** overhead 2. **Minimize** the **control** overhead

Experiment Environment

- Simulation Environment
  - Modelsim SE 5.8d
  - **Cycle Accurate Simulation, VHDL RTL Coding Style**

- Synthesis Environment
  - XILINX ISE 8.2.03i
  - **Target platform – XILINX ML410 (xc4vfx60-11ff1152) FPGA**

Comparisons of Different Buffer Implementation

- The comparison with conventional design in the **same additional hardware overhead**.
  - NoC area model in [6]

Extended Buffer (EB): 80 bits

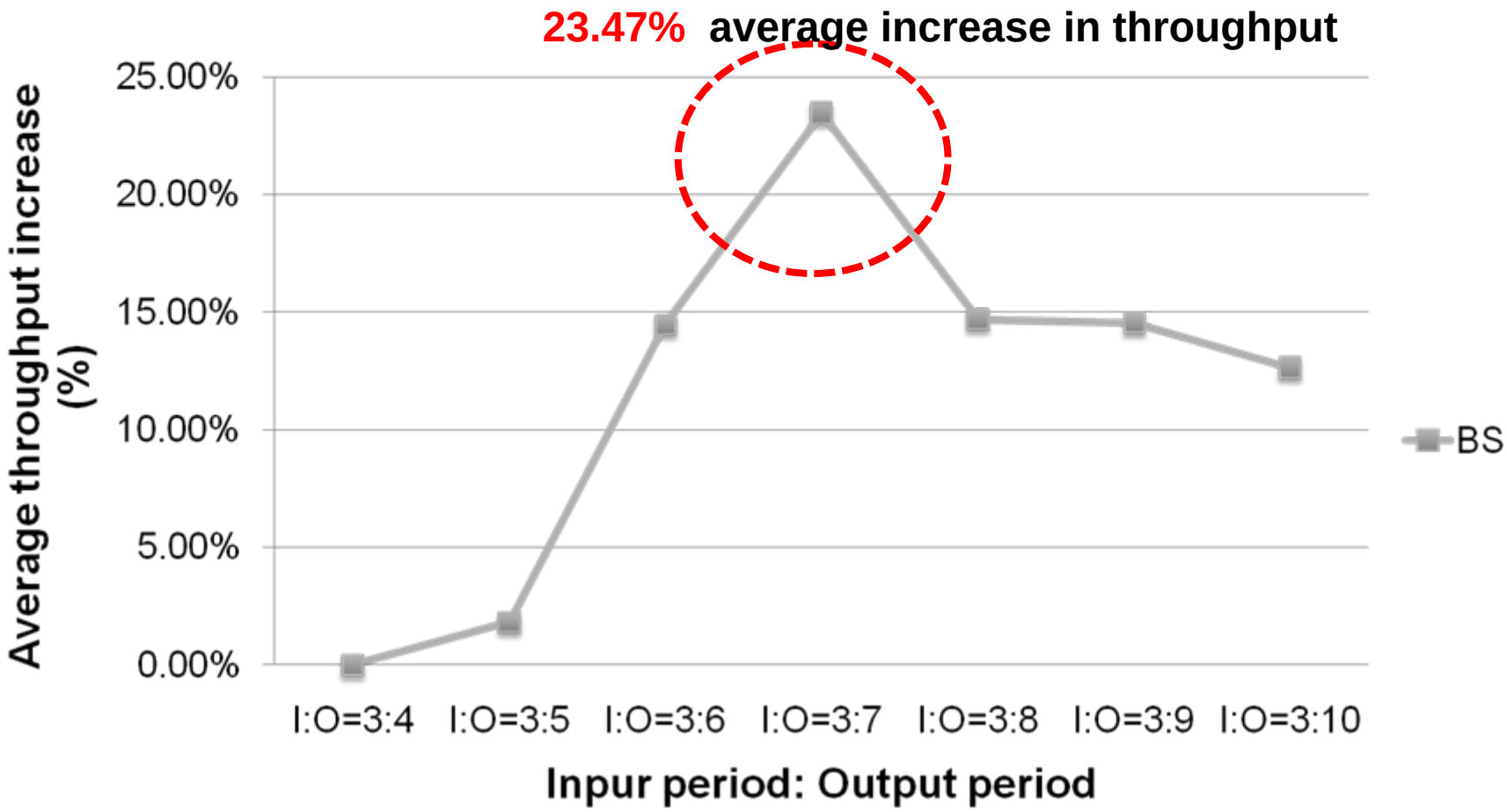
vs.

Buffer Stealing Design: 64 bits

[6] M.-M. Kim and J.-D. Davis and M. Oskin and T. Austin, “Polymorphic On-Chip Networks,” in Proceedings of the 35th International Symposium on Computer Architecture. 2008, pp. 101 -112.

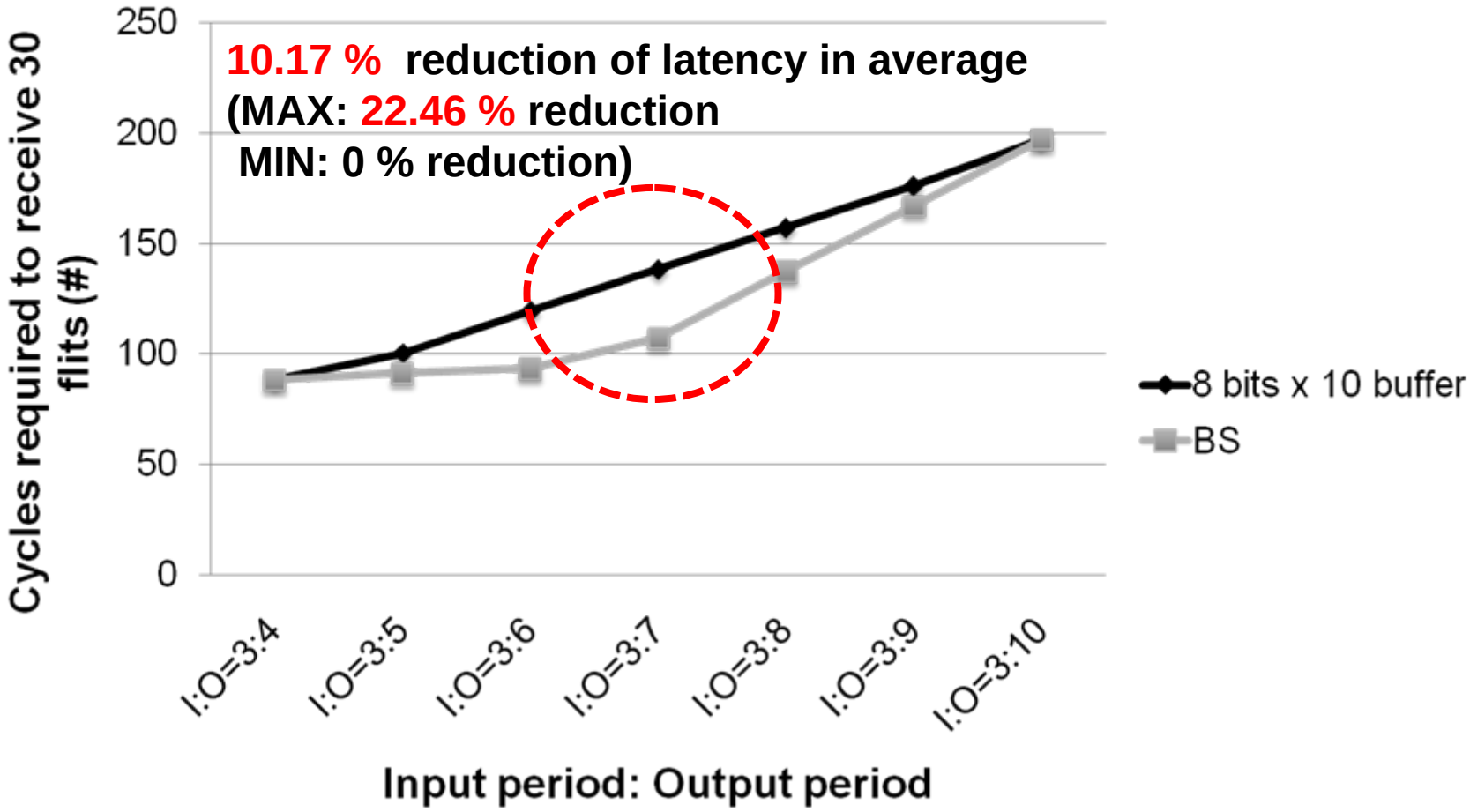
Comparison with Extended Buffer

- Throughput



Comparison with Extended Buffer (cont.)

- Latency

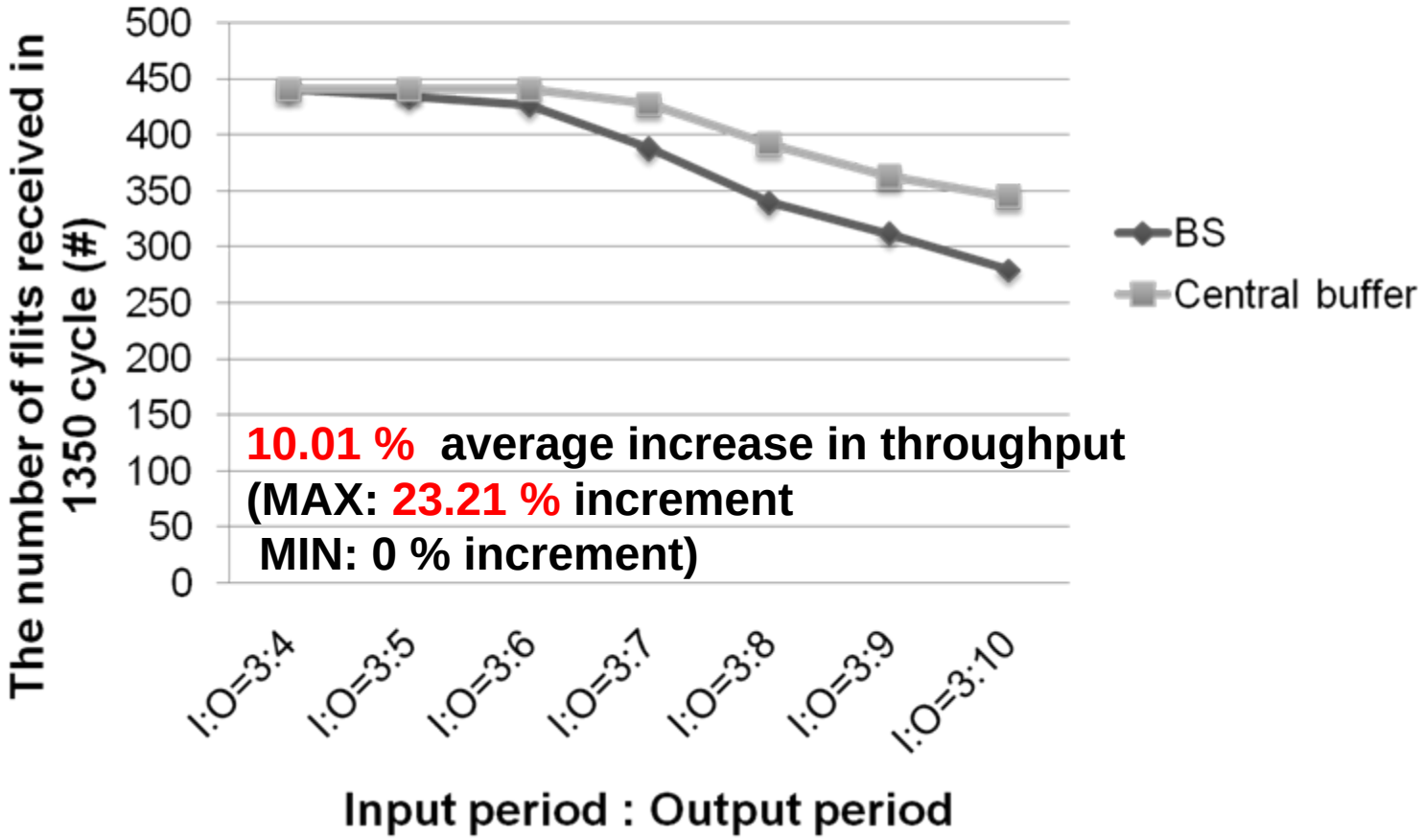


Comparisons of different buffer implementation

- The comparison with **central buffer design**.
  - [18] L.-F. Leung and C.-Y. Tsui, “Optimal link scheduling on improving best-effort and guaranteed services performance in network-on-chip systems,” in Proceedings of the 43rd Design Automation Conference. ACM, July 2006, pp. 833–838.

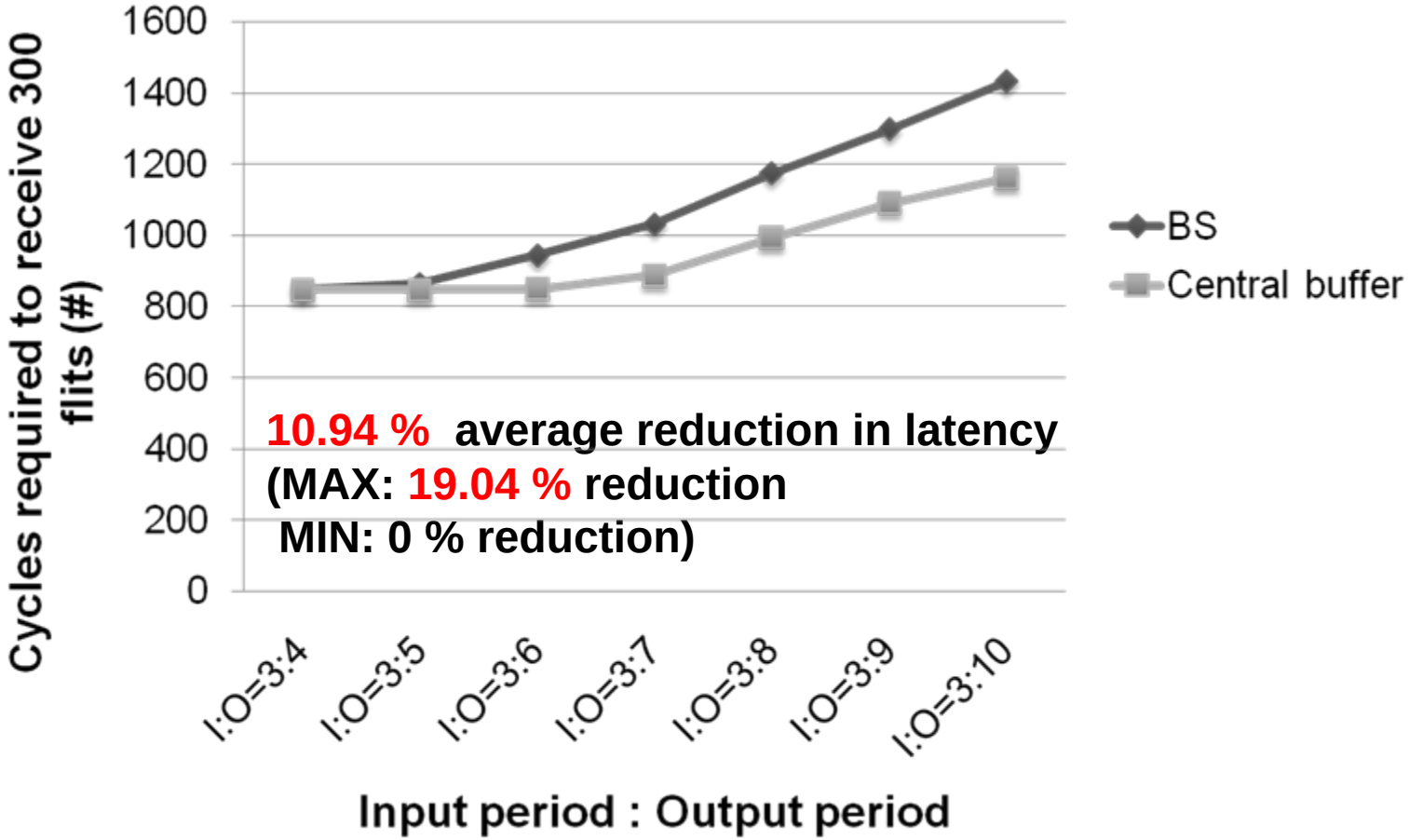
Comparison with Central Buffer

- Throughput



Comparison with Central Buffer (cont.)

- Latency



Comparison with Central Buffer (cont.)

- Performance/ Hardware resource overhead

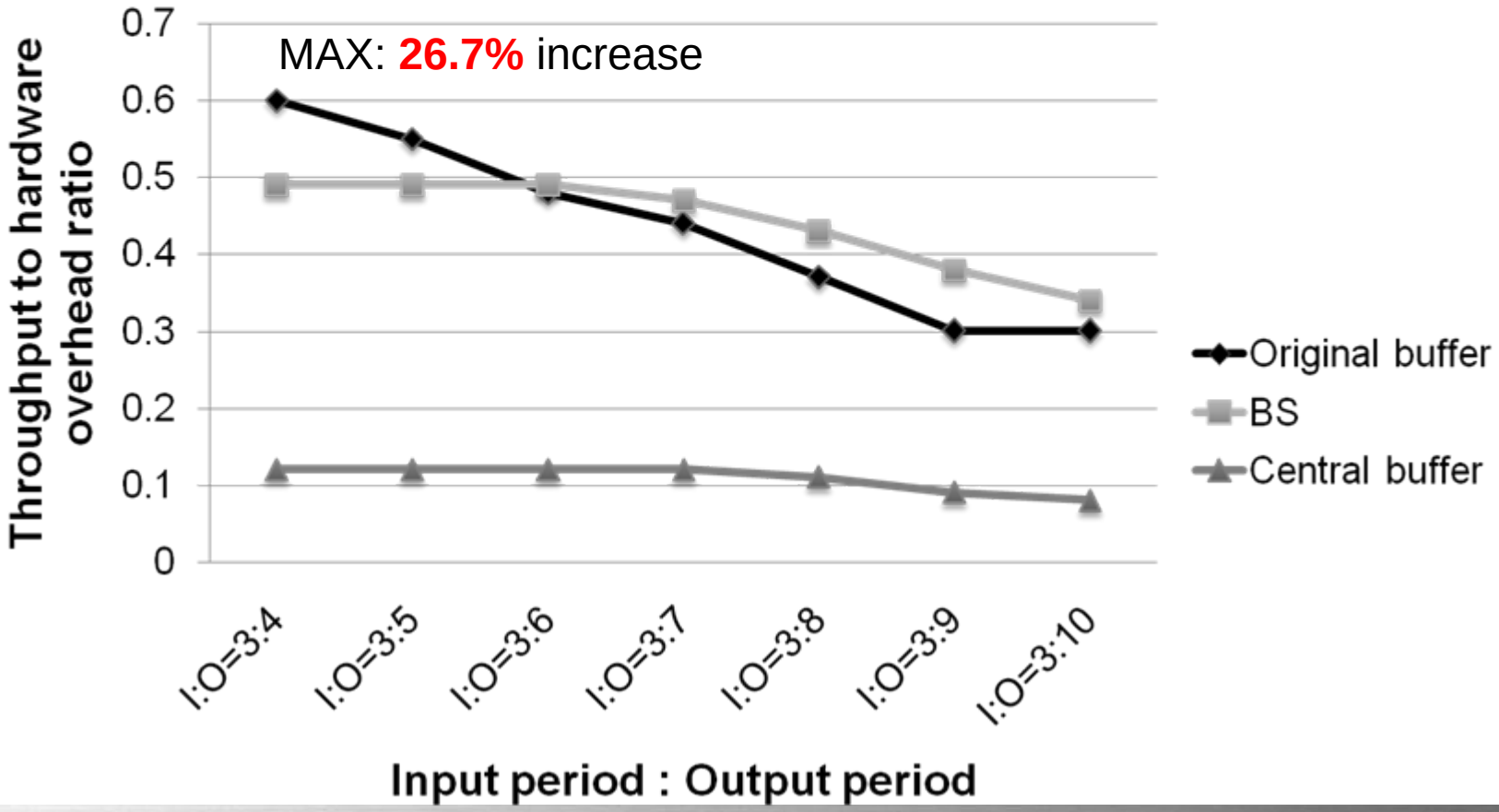
TABLE I
SYNTHESIS RESULTS FOR DIFFERENT BUFFER DESIGNS

Buffer Design	Frequency (MHz)	Hardware overhead (%)
Original buffer	232.056	baseline
Buffer Stealing	203.442	22.18
Central Buffer	142.511	215.48

- It shows that the **large hardware overhead** of the central buffer. (additional 215.5% resources required)
- The central buffer design is not a **cost-efficient** implementation.

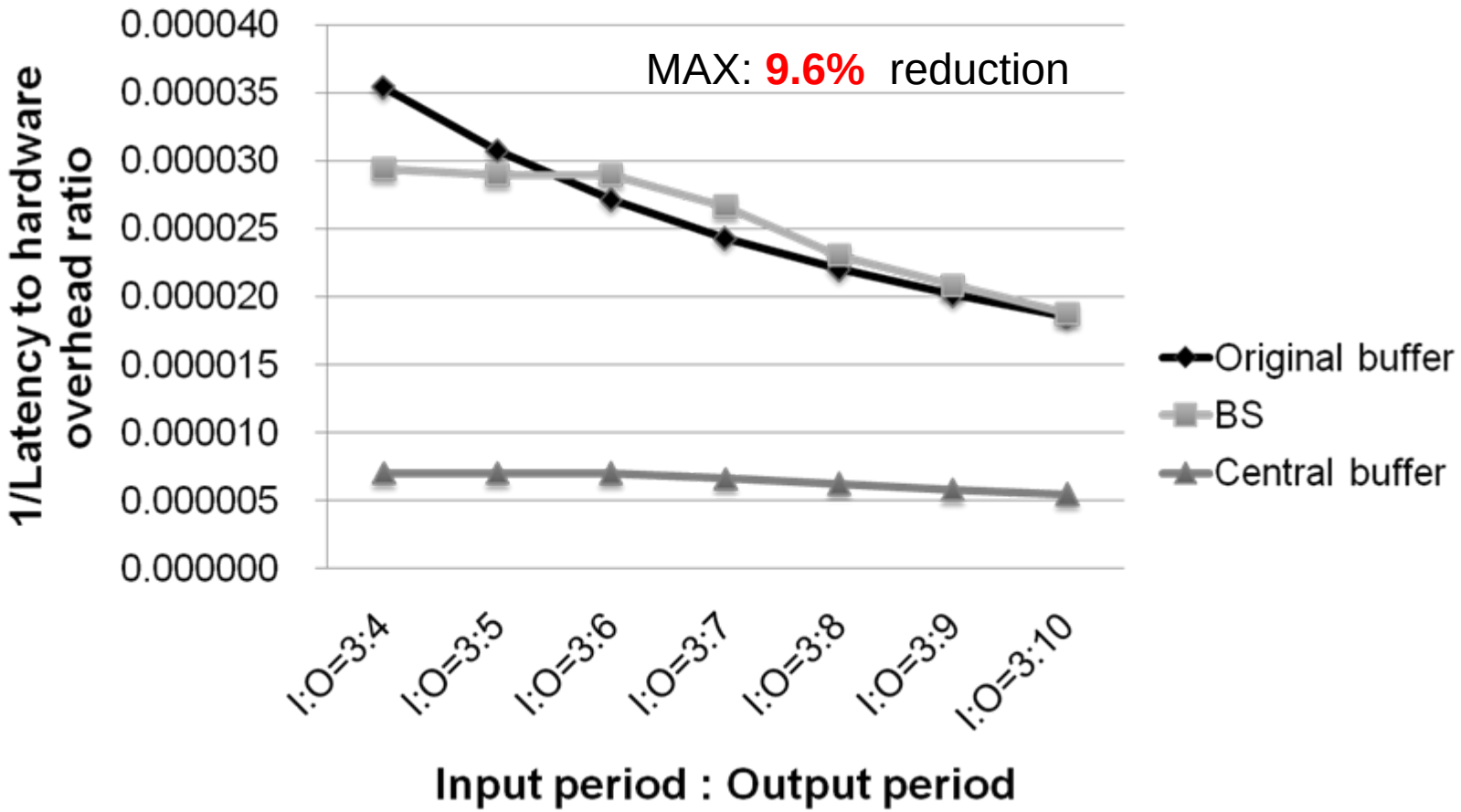
Comparison with Central Buffer (cont.)

- Throughput to hardware overhead ratio (in 1350 cycles)
  - Throughput / hardware overhead (%)



Comparison with Central Buffer (cont.)

- 1/Latency to hardware overhead ratio (300 flits)
  - (1 /Latency) / hardware overhead (%)



Outline

- Introduction
- Motivation and Objective
- Buffer Stealing Design
- Experiments
- **Conclusions**

Conclusions

- The main idea of the proposed **Buffer Stealing (BS) design**

- Experiment results show that
  - **Increase in throughput** of **23.47 % flits** (in average)
  - **Reduce latency** by **10.17 % cycles** (in average)
    - Better congestion toleration

- Future work
  - More **real-world example** implementations
  - The support for **dynamically reconfigurable** system

Thank you for your listening !